BAB III

OBJEK DAN METODE PENELITIAN

3.1 OBJEK PENELITIAN

3.1.1 Sejarah Singkat Perusahaan

INDIGO MOBILE PHONE didirikan oleh bapak taufik pada tahun 2000 bertempat di Bandung Electronic Center (BEC). perusahaan ini bergerak di bidang elektronik yaitu jual beli handphone baru dan bekas.

Dalam dunia bisnis yang semakin kompetitif serta semakin berkembangnya teknologi khususnya teknologi informasi membuat peta persaingan antar perusahaan semakin ketat.

Hal ini sangat menuntut kesiapan bagi manajemen perusahaan tersebut sekaligus menjadi tantangan bagi perusahaan untuk mampu bersaing dalam dunia bisnis yang modern dan menjadi perusahaan yang kompetitif

3.1.2 Visi dan Misi Perusahaan

1. Visi Perusahaan

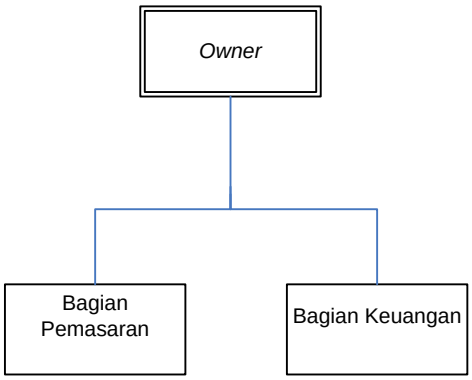
   INDIGO MOBILE PHONE bertujuan untuk menjadi perusahaan handphone yang berkualitas dan memberikan pelayanan terbaik kepada konsumen serta berkompeten dalam persaingan didunia bisnis modern.

2. Misi Perusahaan

a. Bisnis perusahaan tertuju pada penjualan handphone yang disesuaikan dengan permintaan konsumen.

b. Meningkatkan pendapatan perusahaan secara bertahap.

3.1.3 Struktur Organisasi



Gambar 3.1 Struktur Organisasi INDIGO MOBILE PHONE

Sumber: Sejarah INDIGO MOBILE PHONE

3.1.4 Deskripsi Tugas

1. *Owner*.

a. Merencanakan strategi penjualan yang tepat.

b. Membuat keputusan yang tepat untuk mencapai target penjualan.

2. Bagian Pemasaran.

a. Membantu *owner* merencanakan srategi penjualan.

b. Membuat strategi pemasaran yang tepat sehingga dapat menarik konsumen.

3. Bagian Keuangan.

   a. Membuat anggaran pembelian barang untuk dijual kembali.

   b. Mengatur keuangan perusahaan.

3.2 METODE PENELITIAN

3.2.1 Desain Penelitian

Dalam melakukan penelitian, desain penelitian merupakan kerangka atau perincian prosedur kerja yang akan dilakukan pada waktu meneliti, sehingga diharapkan dapat memberikan gambaran dan arah mana yang akan dilakukan dalam melaksanakan penelitian tersebut, desain penelitian yang baik dapat memudahkan kita dalam melakukan penelitian dan sesuai dengan tujuan yang telah ditetapkan sebelumnya.

Sesuai dengan tujuan yang ingin dicapai dalam penelitian ini penulis menggukan metode deskriptif dalam menggumpulkan data sebagai bahan laporan, dengan metode ini penulis melakukan pendekatan atau observasi terhadap sistem yang sedang berjalan di INDIGO MOBILE PHONE agar penulis dapat lebih mudah memperoleh data yang akan dipakai untuk melakukan pengembangan terhadap sistem

Gambaran secara umum desain penelitian adalah sebagai berikut :

1. Observasi

   Yaitu mengidentifikasikan minat penelitian secara umum. Mengamati langsung terhadap bagian-bagian yang diteliti. Hal ini dilakukan untuk mendapatkan data yang tepat.

2. Pengumpulan Data Awal

   Yaitu melakukan wawancara kepada pegawai yang memiliki peranan penting untuk mendapatkan data-data yang akan diteliti dan melakukan survei kepustakaan, yaitu dengan membaca arsip-arsip yang berguna untuk penelitian.

3. Rumusan Masalah

   Merumuskan masalah yang dihadapi perusahaan sebagai bahan perancangan penelitian.

4. Kerangka Teoritis

   Mengidentifikasikan variabel secara jelas.

5. Perumusan Hipotesis
6. Pengumpulan Data dan Analisis
7. Perancangan Perangkat Lunak
8. Pembuatan Perangkat Lunak
9. Penulisan Laporan.

3.2.2 Jenis dan Metode Pengumpulan Data

3.2.2.1 Sumber Data Primer

Data Primer adalah data yang diperoleh langsung dari instansi baik melalui pengamatan maupun pencatatan terhadap objek penelitian, data tersebut di peroleh dengan teknik sebagai berikut:

a. Observasi

Disini penulis melakukan observasi ke seluruh bagian dari perusahaan INDIGO MOBILE PHONE untuk mendapatkan data atau informasi tentang sistem pemasaran dan penjualan yang diterapkan di INDIGO MOBILE PHONE.

b. Interview

Penulis melakukan *interview* atau wawancara ke bagian pemasaran untuk mengetahui bagaimana sistem pemasaran yang ada, lalu ke bagian keuangan untuk mengetahui bagaimana sistem pembukuan dari perusahaan.

3.2.2.2. Sumber Data Sekunder

Skunder adalah data yang diambil secara tidak langsung dari objek penelitian. Data ini diperoleh dari buku-buku, dokumentasi, dan literatur - literatur yang ada relevansinya. Pada saat melakukan penelitian ada beberapa data yang diperoleh dari perusahaan antara lain, jenis data sekunder juga dibagi menjadi dua bagian yaitu:

1. Internal, pedoman kerja, membaca catatan atau arsip-arsip perusahaan, dokumen-dokumen yang berhubungan dengan pemasaran dan penjualan produk dan jasa.

2. Eksternal, membaca buku-buku,hasil-hasil kajian studi kasus atau arsip lainnya yang menyediakan informasi yang cukup banyak mengenai informasi untuk keperluan penelitian dan pemecahan masalah, dan internet.

3.2.3 Metode Pengembangan Dan Pendekatan Sistem

Adapun metode pendekatan dan pengembangan sistem yang digunakan oleh penulis, yaitu:

3.2.3.1 Metode pendekatan Sistem

Metode pendekatan sistem yang digunakan oleh penulis adalah Analisis dan Perancangan Terstruktur.

Tahapan-tahapannya terdiri dari:

1. Flowmap

   Yaitu bagan alir sistem yang digunakan untuk menggambarkan arus dari dokumen-dokumen yang ada di perusahaan atau organisasi.

2. Diagram Kontek (*Contex Diagram*)

   Diagram Kontek adalah diagram tingkat tinggi yang menggambarkan

hubungan antar entitas eksternal dengan sistem, dimana data yang diinputkan oleh bagian komponen eksternal yang akan diproses di dalam sistem dan akan menghasilkan laporan yang akan diinginkan oleh komponen eksternal tersebut.

3. *Data Flow Diagram* (DFD)

   Data Flow Diagram (DFD) adalah sebuah diagram yang sering digunakan untuk menggambarkan secara logika bagaimana data itu mengalir, dimana data tersebut akan disimpan dan kemana saja laporan yang akan dibuat itu diberikan. *Data Flow Diagram* (DFD) ini juga menggambarkan arus data secara terstruktur dari mulai proses input sampai dengan pembuatan laporan yang dihasilkan oleh sistem.

4. Kamus Data (*Data Dictionary*)

   Kamus Data adalah daftar organisasi dari semua elemen data yang ada dalam sistem secara lengkap, dengan definisi yang baku. Sehingga user dan analisis sistem akan memiliki pengertian sama untuk *input*, *output*, komponen penyimpanan serta perhitungannya.

5. Normalisasi

   Salah satu cara pendekatan atau teknik yang digunakan dalam membangun disain *logic* basis data *relation* dengan menerapkan sejumlah aturan dan kriteria standar. Tujuan dari

normalisasi adalah untuk menghasilkan struktur tabel yang normal dan baik. Teknik normalisasi adalah upaya agar disain *logic* tabel-tabel berada dalam "*normal form"* yang dapat didefinisikan dengan menggunakan ketergantungan fungsi.

6. Relasi Tabel

   Relasi tabel merupakan gambaran tentang hubungan antara tabel satu dengan tabel yang lainnya yang ada di dalam suatu sistem.

7. *Entity Relationship Diagram* (ERD)

   Entity Relationship Diagram adalah sebuah diagram yang menggambarkan model relasi antar rancangan data tersimpan (*file*) atau bentuk logika yang dipakai analisis dan desain suatu sistem informasi. Model relasi ini diperlukan untuk menggambarkan struktur data dan relasi antar data.

3.2.3.2 Metode Pengembangan Sistem

Metode pengembangan sistem informasi yang digunakan adalah metode pengembangan sistem secara *prototyping. Prototyping* merupakan suatu pendekatan yang membuat model yang memperlihatkan fitur-fitur produk, layanan, atau sistem usulan. Modelnya dikenal dengan *prototype*. Langkah-langkahnya adalah sebagai berikut:

1. Mengidentifikasi kebutuhan-kebutuhan dasar sistem. Dalam hal ini perancangan sistem bekerja dengan pemakai (*User)* untuk menangkap dasar yang diperlukan pemakai (*User)*
2. Membuat *prototype* dari identifikasi kebutuhan-kebutuhan dasar sistem sesuai dengan apa yang *User* inginkan dan yang orang yang membuat *prototype* sarankan.
3. Menggunakan *prototype*. Pada tahap ini, pemakai *(User)* diminta untuk bekerja dengan sistem untuk menentukan cocok tidaknya *prototype* terhadap kebutuhan pemakai *(User)* dan diharapkan pemakai *(User)* memberi masukan untuk memperbaiki *prototype.*
4. Memperbaiki dan meningkatkan *prototype*. *Prototype* diperbaiki sesuai dengan semua perubahan uang diminta oleh pemakai *(User).* Setelah itu, langkah 3 dan langkah 4 dilakukan secara terus menerus sampai pemakai *(User)* merasa cocok dan puas.

Kelebihan dari metode pengembangan *prototype* adalah pembuat dapat melibatkan pemakai *(User)* dalam pembuatan sebuah sistem informasi yang cocok dan sesuai dengan yang *user* inginkan.

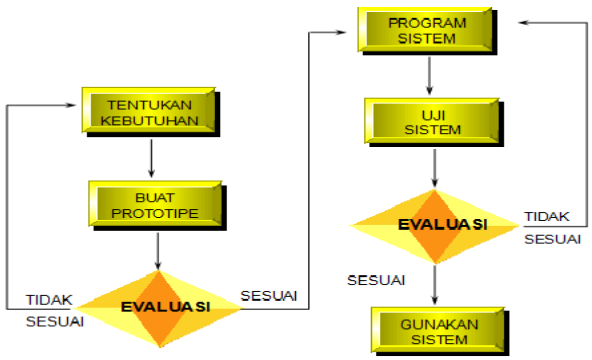
Kekurangan dari metode *prototyping* adalah sebagai berikut:

1. Membuat proses analisis dan perancangan menjadi terlalu singkat. Analisis sistem informasi dalam hal ini mungkin saja

kurang memahami masalah-masalah dan kebutuhan-kebutuhan yang sebenarnya dari perusahaan.

2. Mengesampingkan alternatif pemecahan masalah. Analisis sistem informasi tidak membuat alternatif lain ketika *prototype* pertama yang disajikan mendapat reaksi positif dari *user*.
3. Penerapan system informasi dengan menggunakan metode *prototyping* biasanya kurang fleksibel dalam menghadapi perubahan-perubahan.
4. Tidak selamanya *prototype* yang dihasilkan oleh metode *prototyping* mudah untuk dirubah.
5. Umumnya*prototype* ini terlalu cepat selesai.

Berikut adalah gambar dari metode *prototype*:



Gambar 3.2 metode prototype

Sumber:

http://3.bp.blogspot.com/_nxpJ3EBTF28/S9f5a1maOpI/AAAAAAAAAA0/v4PnO9ipitI/s1600/prototipe.png

3.2.3.3.Alat Bantu Analisis dan Perancangan

Di bawah ini akan dijelaskan mengenai alat bantu analisis dan perancangan yang digunakan oleh penulis

1. Flow Map

Berikut ini adalah penjelasan tentang flowmap yang dikutip dari : http://fizzulhaq.blogspot.com/2009/11/pengertian-sistem-informasi-manajemen.html. *"FlowMap merupakan bagan yang menerangkan bagaimana data dokumen mengalir dari satu bagian kebagian lain, setelah melalui suatu proses pengolahan data".*

Flow Map adalah aliran data berbentuk dokumen atau formulir didalam suatu sistem informasi yang merupkan suatu aktivitas yang saling terkait dalam hubungan dengan kebutuhan data informasi. Diagram aliran dokumen merupakan bagan-bagan alir yang menunjukan arus dari laporan dan formulir termasuk tembusan-tembusan nya.Kegunaan dari flow map ini adalah :

a. Menggambarkan aktivitas apa yang sedang berjalan.
b. Menggambarkan aliran dokumen yang terlihat.
c. Menjelaskan hubungan-hubungan data dan informasi dengan bagian-bagian dalam aktivitas tersebut.

2. Diagram Kontek

Menutut Andri Kristanto, "Perancangan Sistem Informasi dan Aplikasi", Gava Media 2003.

"*Diagram Konteks adalah sebuah diagram sederhana yang menggambarkan hubungan antara entity luar, masukan dan keluaran dari sistem. Diagram Konteks direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem*".

Pengertian Diagram Konteks menurut buku Dikutip dari buku Al-bahra.2005.Analisis dan Desain Sistem Informasi.Graha Ilmu.Yogyakarta.

"Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. DC merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke system atau output dari system, ia kan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh Boundary (dapat digambarkan dengan garis putus).Dalam DC hanya ada satu proses. tidak boleh ada storage dalam diagram konteks".

3. Data Flow Diagram

Menurut Andri Kristianto (2007:61), Data Flow Diagram yaitu

:"*Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dari interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut*".

DFD menggambarkan penyimpanan data dan proses yang mentransformasikan data. DFD menunjukkan hubungan antara data pada sistem dan proses pada sistem. Ada dua teknik dasar DFD yang umum dipakai yaitu Gane and Sarson dan Yourdon dan De Marco".

Dalam mengembangkan suatu aliran data atau proses yang terjadi didalam sistem *data flow diagram* menggunakan simbol-simbol yang memiliki arti tersendiri dalam menerangkan :

A. DFD Leveled

Model ini digambarkan sistem sesbagai jaringan kerja antar fungsi yang berhubungan satu dengan lainnya dengan aliran dan penyimpanan data. Sebagai alat Bantu desain sistem, model ini hanya memodelkan sistem dari pandang fungsi. Dalam DFD leveled ini akn terjadi penurunan level, level yang lebih rendah harus mampu merepresentasikan proses tersebut kedalam spesifikasi proses yng jelas. Jadi dalam DFD *leveled* bisa dimuli dengan level 0 kemudian turun ke DFD level 1 dan seterusnya.

4. Kamus Data

Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga *user* dan analis sistem mempunyai pengertian yang sama tentang *input*, *output*, dan komponen *data strore*.

Kamus data ini sangat membantu analis sistem dalam mendefinisikan data yang mengalir di dalam sistem, sehingga pendefinisian data itu dapat dilakukan dengan lengkap dan terstruktur. Pembentukan kamus data dilaksanakan dalam tahap analisis dan perancangan suatu sistem.

Pada tahap analisis, kamus data merupakan alat komunikasi antara user dan analis sistem tentang data yang mengalir di dalam sistem, yaitu

tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh *user*. Sementara itu, pada tahap perancangan sistem kamus data digunakan untuk merancang input, laporan dan *database*. Pembentukan kamus data didasarkan atas alur data yang terdapat pada DFD. Alur data pada DFD ini bersifat global, dalam arti hanya menunjukan nama alur datanya tanpa menunjukan struktur dari alur data itu. Untuk menunjukan struktur dari alur data secara terinci maka dibentuklah kamus data yang didasarkan pada alur data di dalam DFD dikutip dari http://iaprima.staff.gunadarma.ac.id/Pendekatan.pdf/12 mei 2007/.

4. Perancangan Basis Data

Perancangan merupakan suatu hal yang sangat penting dalam pembuatan basis data. Permasalahan yang dihadapi pada waktu perancangan yaitu bagaimana basis data yang akan dibangun ini dapat memenuhi kebutuhan saat ini dan masa yang akan datang. Untuk itu diperlukan perancangan basis data baik secara fisik maupun secara konseptualnya.

Perancangan model konseptual basis data dalam sebuah organisasi menjadi tugas dari *Administrator* basis data. Model konseptual merupakan kombinasi beberapa cara untuk memproses data untuk beberapa aplikasi. Model konseptual tidak tergantung pada aplikasi individual, DBMS digunakan, *Hardware* komputer dan model fisiknya. Pada perancangan model konseptual basis data ini penekanan dilakukan pada struktur data dan relasi antara file. Pada perancangan model konseptual ini dapat

dilakukan dengan menggunakan model data relasional. Teknik Model Data Realsional ada 2 yaitu:

a. Normalisasi

Kutipan buku Al-bahra.2005.Analisis dan Desain Sistem Informasi.Graha Ilmu.Yogyakarta.

"*Definisi Normalisasi adalah proses pengelompokan data ke dalam bentuk tabel atau relasi atau file untuk menyatakan entitas dan hubungan mereka sehingga terwujud satu bentuk database yang mudah untuk dimodifikasi"*.

Jadi dapat disimpulkan dari peryataan diatas proses Normalisasi, merupakan proses pengelompokan data elemen menjadi tabel-tabel yang menunjukkan entitas dan relasinya. Pada proses normalisasi selalu diuji pada beberapa kondisi, apakah ada kesulitan pada saat menambah, menghapus, mengubah, dan membaca pada suatu *database*.

Bila ada kesulitan pada pengujian tersebut, maka relasi tersebut dipecahkan menjadi beberapa table lagi, sehingga diperoleh *database* yang optimal.Secara umum, proses normalisasi terdiri dalam beberpa tahap , yaitu :

1. Tahap tidak normal

   Pada tahap ini, semua data yang ada direkam tanpa format tertentu. Hal tersebut dapat menyebabkan data mengalami duplikasi.

2. Normalisasi tahap 1

Normalisasi tahap 1 menghilangkan duplikasi data yang terjadi pada tahap tidak normal dengan cara menghapus data-data yang sama.

3. Normalisasi tahap 2

   Adalah menentukan kunci dari normalisasi tahap 1 yang akan digunakan sebagai *primary key* pada tabel, membentuk tabel berdasarkan *primary key* dan mengelompokan data pada tabel-tabel yang sudah dibentuk.

4. Normalisasi tahap 3

   Pada tahap 3 dilakukan penentuan relasi antar tabel sehingga memungkinkan adanya field kunci sekunder.

b. Tabel Relasi atau ERD

Dikutip dari fathansyah buku teks komputer basis data. Informatika.tahun 2007.

" ERD adalah Model entity relationship yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakata dari dunia nyata yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan diagram entity relationship (diagram E-R) ".

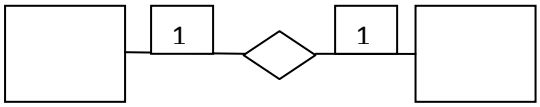
ERD merupakan model konseptual yang mendiskripsikan hubungan antara penyimpana, ERD digunakan untuk memodelkan struktur data dan hubungan antar data, pengguna dapat menguji model dengan mengabaikan proses yang harus dilakukan.

Ada beberapa element yang terdapat dalam ERD, diataranya :

1. *Entity* adalah sesuatu yang ada dalam sistem nyata maupun abstrak dimana data tersimpan atau dimana data itu berada. Entitas digambarkan dalam bentuk persegi panjang dan diberi nama benda.
2. *Relationship* adalah hubungan alamiah yang terjadi anyar entitas. Pada umum nya penghubung atau *realitionship* diberi nama dengan kata kunci dasar sehingga memudahkan pembacaan relasi. *Realitionship* dapat di gambarkan dalam bentuk ketupat
3. *Relationship Degree* adalah jumlah entitas yang berkaitan dalam suatu relasi.
4. Atribut adalah sifat atau karakteristik dari setiap entitas dari realitionship. Pada intinya atribut menjelaskan apa yang dimaksud entits. Ada beberapa macam kardinalitas yaitu:
    a. Satu ke satu *(One to One)*

        Satu ke satu (*One to One)* adalah hubungan antara satu atribut dengan atribut lain dalam satu *file* yang sama mempunyai hubungan satu lawan satu.

Contoh :



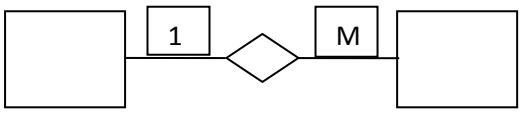
Gambar 3.4 Relationship one to one

(Sumber Ir. Fathansyah, Basis Data, Inforamatika Bandung, 2000)

b. Satu ke banyak *(One to Many*)

Satu ke banyak *(One to Many)* adalah hubungan antara atribut yang satu dengan atribut laing dalam satu *file* yang sama mempunyai hubungan satu lawan banyak.

Contoh:



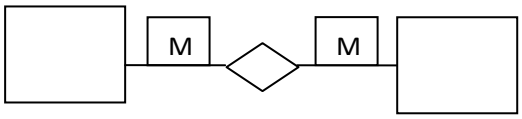
Gambar 3.5 Relationship one to many

(Sumber Ir. Fathansyah, Basis Data, Inforamatika Bandung, 2000)

c. Banyak ke banyak *(Many to Many)*

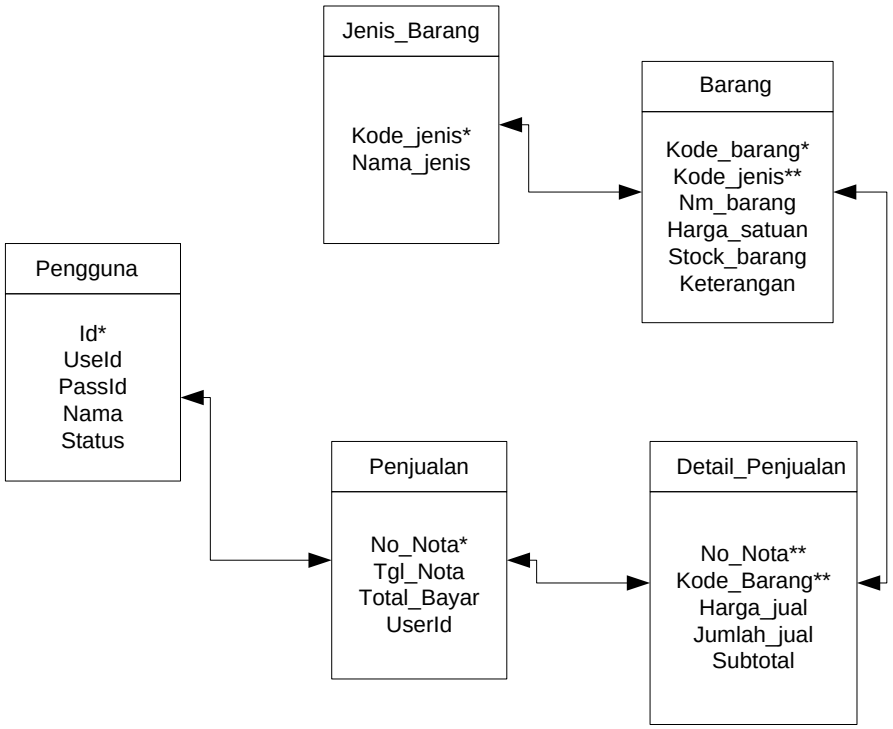
Banyak ke banyak *(Many to Many)* adalah hubungan antara atribut yang satu dengan atribut lain dalam satu *file* yang sama mempunyai hubungan banyak lawan banyak.

Contoh :



Gambar 3.6 Relationship many to many

(Sumber Ir. Fathansyah, Basis Data, Inforamatika Bandung, 2000)



Gambar 3.7 Tabel Relasi atau ERD penjualan

3.2.4. Pengujian Software

Dikutip dari *"Ayuliana/Testing dan Implementasi/Mar2009/"* Ujicoba *software* merupakan elemen yang kritis dari SQA dan merepresentasikan tinjauan ulang yang menyeluruh terhadap spesifikasi,desain dan pengkodean. Ujicoba merepresentasikan ketidaknormalan yang terjadi pada pengembangan *software*. Selama definisi awal dan fase pembangunan, pengembang berusaha untuk membangun *software* dari konsep yang abstrak sampai dengan implementasi yang memungkinkan. Strategi uji coba mempunyai karakteristik sebagai berikut :

1) Pengujian mulai pada tingkat modul yang paling bawah, dilanjutkan dengan modul di atasnya kemudian hasilnya dipadukan.
2) Teknik pengujian yang berbeda mungkin menghasilkan sedikit perbedaan (dalam hal waktu).
3) Pengujian dilakukan oleh pengembang *software* dan (untuk proyek yang besar) suatu kelompok pengujian yang independen.
4) Pengujian dan debugging merupakan aktivitas yang berbeda, tetapi *debugging* termasuk dalam strategi pengujian.

Pengujian software adalah satu elemen dari topik yang lebih luas yang sering diacu sebagai *verifikasi dan validasi.*

1. *Verifikasi* : Kumpulan aktifitas yang menjamin penerapan *software* benarbenar sesuai dengan fungsinya.
2. *Validasi* : Kumpulan aktivitas yang berbeda yang memastikan bahwa *software* yang dibangun dapat memenuhi keperluan pelanggan.

Selama beberapa dekade, metode desain ujicoba kasus telah dikembangkan. Metode ini menyediakan pendekatan sistematik untuk ujicoba. Hal yang lebih penting yaitu, metode-metode ini menyediakan mekanisme yang dapat membantu memastikan kelengkapan ujicoba dan menyediakan tingkat kemungkinan yang tinggi dalam penemuan kesalahan pada *software.* Semua produk yang dikembangkan (*engineered*) dapat diujicoba dengan salah satu cara dari dua cara berikut :

1. Mengetahui fungsi-fungsi yang dispesifikasikan pada produk yang didesain untuk melakukannya, ujicoba dapat dilakukan dengan mendemonstrasikan setiap fungsi secara menyeluruh.
2. Mengetahui cara kerja internal dari produk, ujicoba dapat dilakukan untuk memastikan bahwa seluruh operasi internal dari produk dilaksanakan berdasarkan pada spesifikasi dan komponen internal telah digunakan secara tepat.

Pendekatan pertama adalah *black box testing* dan yang kedua adalah *white box testing*. Menurut (Pressman 2002:551) yang dikutip dari *"Ayuliana/Testing dan Implementasi/Mar2009/"*.

"Black box testing Pengujian black-box sering disebut juga dengan pengujian tingkah laku (behavoiur testing) yang lebih terfokus kepada kebutuhan fungsional dari perangkat lunak, pengujian black-box memungkinkan pembuat perangkat lunak untuk menentukan kondisi yang terjadi untuk suatu masukan yang akan menjalankan semua kebutuhan fungsional."

Jadi dapat disimpulkan bahwa *black box* testing adalah pengujian yang dilkukan melalui antar muka perangkat lunak atau *interface* untuk mengetahui apakah kondisi masukan dan keluaran sudah berjalan dengan yang diharapkan.

Pengujian *black-box* dilakukan untuk menentukan beberapa macam kesalahan, yaitu :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau *database external*
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

Dikutip dari *"Ayuliana/Testing dan Implementasi/Mar2009/"* ,*white box testing* didasarkan pada pemeriksaan *detail* prosedural. Alur logikal suatu *software* diujicoba dengan menyediakan kasus ujicoba yang melakukan sekumpulan kondisi dan/atau perulangan tertentu. Status dari program dapat diperiksa pada beberapa titik yang bervariasi untuk menentukan apakah status yang diharapkan atau ditegaskan sesuai dengan status sesungguhnya. Sepintas seolah-olah *white box testing* akan menghasilkan program yang 100 persen benar, yang diperlukan hanyalah mendefinisikan alur logikal, membangun kasus uji untuk memeriksa *software* tersebut dan mengevaluasi hasil yang diperoleh.

Sayangnya, ujicoba yang menyeluruh ini menghadirkan masalah logical tertentu. Untuk sebuah program sederhana sekalipun, terdapat banyak alur logikal yang memungkinkan. Sehingga *white box testing* sebaiknya hanya dilakukan pada

alur logikal yang penting. Struktur data-struktur data yang penting dapat diujikan dengan uji validitas. Atribut dari *black box testing* dan *white box testing* dapat dikombinasikan untuk digunakan bersama.